Attorney Docket: 042503-0261928
Client Reference: ET-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
Board of Patent Appeals and Interferences

In re application of: DENNIS L MONTGOMERY

Application No.: 09/727,096 Group No.: 2165

Filed: November 29, 2000 Examiner: T. Mahmoudi

Title: METHOD AND APPARATUS FOR ENCODING INFORMATION USING MULTIPLE PASSES AND DECODING IN A SINGLE PASS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S BRIEF (37 C.F.R. section 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on April 26, 2006. The fees required under Section 1.17(c), and any required petition for extension of time for filing this brief and fees therefor, are dealt with in the accompanying TRANSMITTAL.

I. REAL PARTIES IN INTEREST (37 C.F.R. section 1.192(c)(1))

The real party in interest in this appeal is the following party: Etreppid Technologies LLC.

II. RELATED APPEALS AND INTERFERENCES (37 C.F.R. section 1.192(c)(2))

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are none.

III. STATUS OF CLAIMS (37 C.F.R. section 1.192(c)(3))

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

There are 18 claims in the application: 2-4, 6-7, 11, 14, 16-22, 26, 29, 47 and 54

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 1, 5, 8-10, 12-13, 15, 23-25, 27-28, 30-46, 48-53, 67-73.
2. Claims withdrawn from consideration but not canceled: None.
3. Claims pending: 2-4, 6-7, 11, 14, 16-22, 26, 29, 47 and 54.
4. Claims allowed: 22 and 26.
5. Claims rejected: 2-4, 6-7, 11, 14, 16-21, 29, 47 and 54.

C. CLAIMS ON APPEAL

The claims on appeal are: 2-4, 6-7, 11, 14, 16-21, 29, 47 and 54.

IV. STATUS OF AMENDMENTS

Applicant filed an Amendment After Final dated April 26, 2006. The one amendment to dependent claim 20 was not entered, but the arguments regarding the independent claim 17 were considered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 17 is directed to a method of further compressing already compressed blocks, using a plurality of first threads for the first compression operations, and a plurality of second threads for the second compression operations. During the compression process, the data is operated upon such that a compression algorithm repeatedly, in a cyclical manner, compresses data that in a previous pass was already compressed by the compression engine. *Id. at page 3, line 20.* The present invention can operate independently on each block. *Id. at page 12, line 23.* Thus, the interface controller needs to be able to determine when to create a new thread or when to use the same thread for multiple blocks. Once the thread determination is complete, the appropriate control signals and blocks of data are transmitt ed to the C/D engine 230, so that the compression of each of the blocks within a give thread can take place and the first pass of compression is initiated. *Id. at page 14, lines 11-13.*

After the result of the first pass of the compression routine on the initial block or blocks the results, which may be interim, are obtained, and stored in buffer manager. *Id. at page 16, line 5.* After completion of the first pass compression by the C/D engine and removal of the threads associated with the first pass compression, the compressed data is associated with second threads, and thereafter the compressed data is operated upon at the initiation of the second pass using the seconds threads. Once further compression using the second threads are completed, the second threads are removed, preserving the second blocks that have then been twice compressed. *Id. at page 16, line 10; page 17, line 13; page 19, line 6.*

At the end of a successful compression routine for a stream of data, a series of compressed blocks will result. *Id. at page 21, lines 4-21.* The decompression routine, however can perform the inverse decompression operation in a single pass. As a result, while compression may require several compression passes on the same data, decompression can always be achieved in a single pass, since the original data stream can be derived from finally compressed data. *Id. at page 21, lines 9-21.*

VI. GROUNDS OF REJECTIONS TO BE REVIEWED ON APPEAL

Claims 2-3, 6-7, 16-18, and 20-21 and 47 are rejected under as being patentable in view of Ageenko etal (IEEE pp 757-761, 1999) in view of US Patent No. 6,366,289 to Johns.

Claims 4, 11, 19 and 54 are rejected under as being patentable in view of Ageenko etal (IEEE pp 757-761, 1999) in view of US Patent No. 6,366,289 to Johns and further in view of US Patent No. 5,586,280 to Simms.

Claim 14 is rejected under as being patentable in view of Ageenko etal (IEEE pp 757-761, 1999) in view of US Patent No. 6,366,289 to Johns and further in view of US Patent No. 6,043,097 to Morikawa.

Independent claim 17 contains allowable subject matter and should be separately considered, along with all other claims dependent thereon, including those claims that deserving of separate consideration as identified below.

Claim 20 contains allowable subject matter and should be separately considered.

Claims 6 and 21 contain allowable subject matter and should be considered together.

Claims 4, 19, and 54 contain allowable subject matter and should be considered together.

Claim 7 contains allowable subject matter and should be separately considered.

Claim 11 contains allowable subject matter and should be separately considered.

Claim 14 contains allowable subject matter and should be separately considered.

VII. ARGUMENTS

Claim 17

Claim 17 is directed to a method of further compressing already compressed blocks. This is not taught or suggested by Ageenko. Different clusters are each compressed once, as in typical in the art. There is not, however, any teaching whatsoever of compressing blocks, and then further compressing the already compressed blocks again. As such, Ageenko does not teach or suggest this element.

In the advisory action dated May 22, 2006, the Examiner for the first time explained his reasoning why Ageenko allegedly teaches this aspect of the present invention. The Examiner refers to Section 3.1 of Ageenko for the first compression steps, and Section 3.2 of Aggenko in support of the second compression steps, in which it is recited that already compressed blocks are further compresses.

A careful reading of the Aggenko article shows that the Examiner's interpretation of Ageenko is wrong. Section 3.1 is for "Model Construction, as shown in the highlighted heading of the Article. What is meant by "model construction is identified previously in Ageenko, in the discussion of the "Forward Adaptive Technique" that is in Section 3 of Ageenko. Ageenko teaches:

> "The model is constructed from statistics gathered over the whole image and stored in the compressed file. The model is used for re-initializing the probability estimation pointers when the compression of a cluster starts." (Ageenko, §3, p. 758-759).

Clearly, the model is used not for compression, but for another purpose. As explicitly taught above, the model is used to re-initialize pointers "<u>when the compression of a cluster starts."</u> There is no teaching or suggestion that a cluster is twice compressed. And Ageenko in Section 3.2 reaffirms this, teaching "[I]n the second stage, the clusters are compressed separately by the QM-coder."

Applicant notes, respectfully, that with respect to the Examiner's reference in Section 3.1 to a "compressed file," the teaching here is that the <u>model table</u> that is stored at the beginning of a compressed file (which compressed file is the file that results from the compression that is performed in

Section 3.2). This model table is <u>not</u> that data from the clusters that is being compressed. Rather, the model contains probabilities that allow for the "fast-attack states," which the "forward adaptive technique of Ageenko uses to speed up the overall time that is used for compression. While there may be advantages to this approach, the approach is not that of the present invention.

Since Ageenko does not teach of suggest anything other than a conventional technique of compressing a cluster once, it does not teach or suggest the present invention. And Johns, as admitted by the Examiner, does not make up for this deficiency.

<u>Claim 20</u>

Claim 20 highlights the above distinction even more clearly, with the language "during the step of operating upon each of the plurality of second threads, a same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed block."

This is not taught or suggested by Ageenko or Johns, as there isn't even a further compression operation on the same block.

<u>Claims 6 and 21</u>

With respect to claims 6 and 21, these claims require "combining" compressed blocks from the threads after the compression operation using the plurality of threads, as recited. In contrast, the Examiner points to an operation in Johns that creates a linked list of various chunks of data, but these various chunks of data have not been previously separated and associated with various threads, and then operated upon as recited. Thus, claims 6 and 21 contains allowable subject matter.

<u>Claim 7</u>

Claim 7 recites that "the step of creating the plurality of first threads includes the step of associating each of the plurality of blocks of digital data with one of the plurality of first threads." This relates to a distinct association between each thread and a block of data. In contrast, the teaching of Johns, which the Examiner relies on to reject claim 7, specifically states that the VFB controller "computes the address of the compressed block control data associated with the chunk and reads this control data from the memory device where it resides." See Johns, col. 7 at lines 62-66. It is clear that Johns teaches an association between the compressed block control data and the chunk; in other words data is being associated with data. Further, however, this association is being performed *after*

compression has already taken place. In contrast, as recited, in the step of creating the first plurality of threads, data is being associated with different threads *prior* to it being compressed, and as such Johns does not provide teaching that render this claim unpatentable.

Claims 4, 19, 54

Each of claims 4, 19, and 54 state that threads include data and threads are independently operated upon in parallel. And applicant notes that claim 54 requires *both* the first and second threads to be operated upon in parallel -- a suggestion nowhere found in any of these references.

With respect to claims 4, 19 and 54, the Examiner adds Simms to the rejection in addition to Ageenko and Johns. The Examiner admits that Ageenko and Johns do not teach "threads are independently operated upon in parallel."

Additionally, Simms does not teach operating upon threads in parallel. Simms teaches:

> "For data which is being compressed, in parallel with organization of the compressed data into groups, the entity manager EM generates an entity header portion for the current entity, which will contain the compressed record data." (See col. 17, lines 16-20."

Separately and independently, Simms also teaches:

> "In parallel with the generation of the main data blocks, 35-byte sub-data blocks are also generated that contain the sub-codes ...." (See col. 19, lines 27-28.)

While these two passages illustrate various different parallel operations going on, they do not illustrate the claimed operations in parallel -- as they do not even refer to operations that are the same as the claimed operations - nor do they relate to usage of threads as recited.

The Examiner also provides no reason for motivation to combine Simms with the other references other than a conclusory assertion that system performance would be improved as well as more efficient compression would be obtained. These conclusory assertions, however, are not based in fact, as the alleged increases in efficiency and reduction in operation time is wholly speculative and unsupported. The mere fact that such parallel operations could potentially increase efficiency and reduce operation time is not sufficient because such as a conclusion does not explain the increase complexity and solution for potential problems that may cause malfunction. Thus, on the established

record there is no support for the Examiner's assertion that such a motivation would have been part of generally available knowledge as this only takes into account the end result and not how that end result is achieved.

Claim 11

With respect to claim 11, Appellant's claim 11 is not obvious in light of Ageenko and Johns as modified by Simms. Claim 11 teaches the

> "step of creating each of the plurality of first threads uses a data type of each of the plurality of blocks so that each of the first threads contains blocks which have a similar data type."

The Examiner admits that neither Ageenko, Johns nor Simms contain the teaching of claim 11, but asserts that Simms does, explaining in the Office Action that Simms teaches:

> "the step of creating each of the plurality of first threads uses a data type of each of the plurality of blocks so that each of the first threads contains blocks which have a similar data type (see column 7, lines 11-16)."

In short the Examiner has merely recited Appellant's claim language and provide a citation to Simms and concluded that this would have been obvious because it would "enable the system for categorize data into blocks of data with common characteristics among the data items." The cited portion of Simms, however, does not support Examiner's rejection. The cited portion specifically states that

> "entries in the block access table each comprise a FLAG entry indicating the type of the entry and a COUNT entry indicating its value. The FLAG field is 8 bits and the COUNT filed is 24 bits. The bits in the FLAG field have the following significance . . ."

This portion is unrelated to the subject matter of claim 11, and, accordingly, the subject matter of claim 11 is patentably distinct.

Claim 14

With respect to claim 14, the Examiner adds Morikawa to the rejection based upon Ageenko and Johns.

The Examiner asserts that Morikawa teaches "the step of predicting and estimated compression time (see column 2, lines 14-18) and estimated compression amount for each block (see column 5, lines 57-63)" is based upon a misinterpretation of Morikawa.

Initially, Morikawa is combined with Ageenko and Johns in a manner where there would not have been motivation to do so. Morikawa is a specific technique used in a device used to print on paper an electronic document, which is not the concern of Ageenko or Johns at all. As such, one of ordinary skill in the art would not have attempted the combination suggested by the Examiner, and this illustrates that inappropriate hindsight is being used.

Assuming arguendo that the art did teach the modification of Ageenko and Johns by Morikawa, such a combination still does not teach, disclose, or suggest Appellant's invention as set forth in claim 14. In particular, Morikawa, however, measures the compression time for one block in order to estimate the time for compressing all of the blocks, which is not the same as "estimating compression time . . . for each block" as set forth in Appellant's claim 14.

Similarly, Morikawa teaches actually measuring and storing information related to "size of compressed image and measured compression time" without any reference to estimating compression amount. See id, col. 5, lines 45-63. Thus, measuring and storing information related to actual "size of compressed image", as taught by Morikawa, is not the same as "estimating compression amount for each block" as set forth in Appellant's claim 14. Thus, the Examiner's rejection of claim 14 is improper.

CONCLUSION

For the reasons set forth above, Appellant asserts that the current rejections are improper, and respectfully submits that the Final Rejection of October 26, 2005 should be reversed and all claims allowed by this Honorable Board.

Date: November 27, 2006

PILLSBURY WINTHROP SHAW PITTMAN LLP

2475 Hanover Street
Palo Alto, CA 94304-1114
Telephone: (650) 233.4790
Customer Number: 27498

DAVID A. JAKOPIN
Registration No. 32995

VIII. CLAIMS APPENDIX

1. (Cancelled)

2. (Previously Presented) A method according to claim 17 wherein the step of operating upon each of the first threads performs lossless compression.

3. (Previously Presented) A method according to claim 17 wherein the step of operating upon each of the first threads independently operates upon each of the plurality of first threads.

4. (Previously Presented) A method according to claim 17 wherein at least certain ones of the first threads are independently operated upon in parallel.

5. (Cancelled)

6. (Previously Presented) A method according to claim 17 further comprising the step of combining compressed blocks in each of the plurality of compressed first threads to obtain digitally compressed data.

7. (Previously Presented) A method according to claim 17 wherein the step of creating the plurality of first threads includes the step of associating each of the plurality of blocks of digital data with one of the plurality of first threads such that blocks within each of the plurality of first threads share certain common compression characteristics.

8. – 10. (Cancelled)

11. (Previously Presented) A method according to claim 17 wherein the step of creating each of the plurality of first threads uses a data type of each of the plurality of blocks so that each of the first threads contains blocks which have a similar data type.

12. – 13. (Cancelled)

14. (Previously Presented) A method according to claim 17 further including the step of predicting an estimated compression time and estimated compression amount for each block.

15. (Cancelled)

16. (Previously Presented) A method according to claim 17 wherein the step of partitioning data includes the step of determining a size of each of the plurality of blocks taking data type of each block into account.

17. (Previously Presented) A method of operating upon digital data comprising the steps of:

partitioning the digital data into a plurality of blocks;

creating a plurality of first threads, such that each first thread includes at least one of the plurality of blocks; and

operating upon each of the plurality of first threads to obtain a plurality of compressed first threads, each compressed first thread including at least one compressed block of digital data;

operating upon each of the compressed first threads to eliminate each of the compressed first threads and retain the compressed first blocks;

creating a plurality of second threads, such that each second thread includes at least one of the plurality of compressed first blocks; and

operating upon each of the plurality of second threads to obtain a plurality of compressed second threads, each compressed second thread including at least one compressed second block of digital data.

18. (Original) A method according to claim 17 wherein the step of operating upon each of the second threads independently operates upon each of the plurality of second threads.

19. (Original) A method according to claim 17 wherein at least certain ones of the

second threads are independently operated upon in parallel.

20. (Original) A method according to claim 17 wherein, during the step of operating upon each of the plurality of second threads, the same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed block.

21. (Previously Presented) A method according to claim 17 further comprising the step of combining the compressed blocks in each of the plurality of compressed second threads to obtain digitally compressed data.

22. (Original) A method according to claim 17 wherein the step of creating the plurality of second threads includes the step of associating each of the plurality of compressed first blocks with one of the plurality of second threads such that compressed first blocks within each of the plurality of second threads share certain common compression characteristics.

23. – 25. (Cancelled)

26. (Original) A method according to claim 17 wherein the step of operating upon each of the plurality of first threads also results in obtaining a plurality of first metadata sets, each first metadata set including portions of compressed first blocks which are determined to possibly have redundancies disposed therein.

27. – 28. (Cancelled)

29. (Original) A method according to claim 17 wherein each first thread has an associated first metadata set.

30. – 46. (Cancelled)

47. (Previously Presented) A method according to claim 17 wherein each first thread further includes control signals.

48. – 53. (Cancelled)

54. (Previously Presented) A method according to claim 19 wherein at least certain ones of the first threads are independently operated upon in parallel.

55-63. (Cancelled)

64. (Previously Presented) A method according to claim 61, wherein the step of compressing further includes the steps of:

operating upon each of the compressed first threads to eliminate each of the compressed first threads and retain the compressed first blocks;

creating a plurality of second threads, such that each second thread includes at least one of the plurality of compressed first blocks; and

operating upon each of the plurality of second threads to obtain a plurality of compressed second threads, each compressed second thread including at least one compressed second block of digital data.

65. (Previously Presented) A method according to claim 64 wherein at least certain ones of the second threads are independently operated upon in parallel.

66. (Previously Presented) A method according to claim 64 wherein, during the step of operating upon each of the plurality of second threads, the same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed block.

67 – 73. (Cancelled)

APPENDIX: EVIDENCE

None.

APPENDIX: RELATED PROCEEDINGS

None.